

ESP8266 系列常见问题



版本 V1.0

版权 ©2019

关于免责声明和版权公告：

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中所得测试数据均为安信可实验室测试所得，实际结果可能略有差异。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

注意：

由于产品版本升级或其他原因，本手册内容有可能变更。深圳市安信可科技有限公司保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，深圳市安信可科技有限公司尽全力在本手册中提供准确的信息，但是深圳市安信可科技有限公司并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。



关于本文档：

1. 本文档收集安信可 ESP8266 系列的模组的常见问题。
2. 如文档所提出的问题解决方法有误，你可以通过发送邮件到我司的技术支持邮箱 support@aithinker.com，我司会安排相关人员进行修改并审核进行更新
3. 本文档最终解释权归深圳市安信可科技有限公司所有

文件制定/修订/废止履历表

版本	日期	制定/修订内容	制定	核准
V1.0	2019.10.29	首次发布	黄方可	徐宏



目录

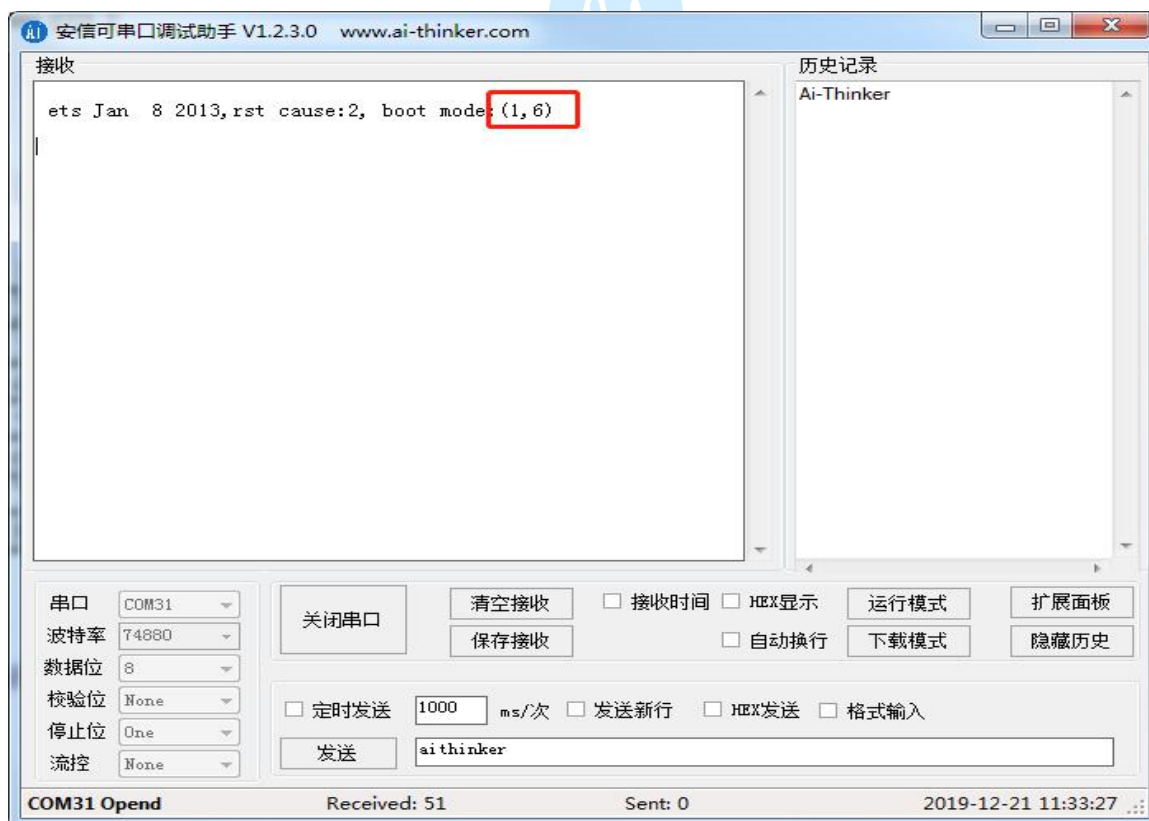
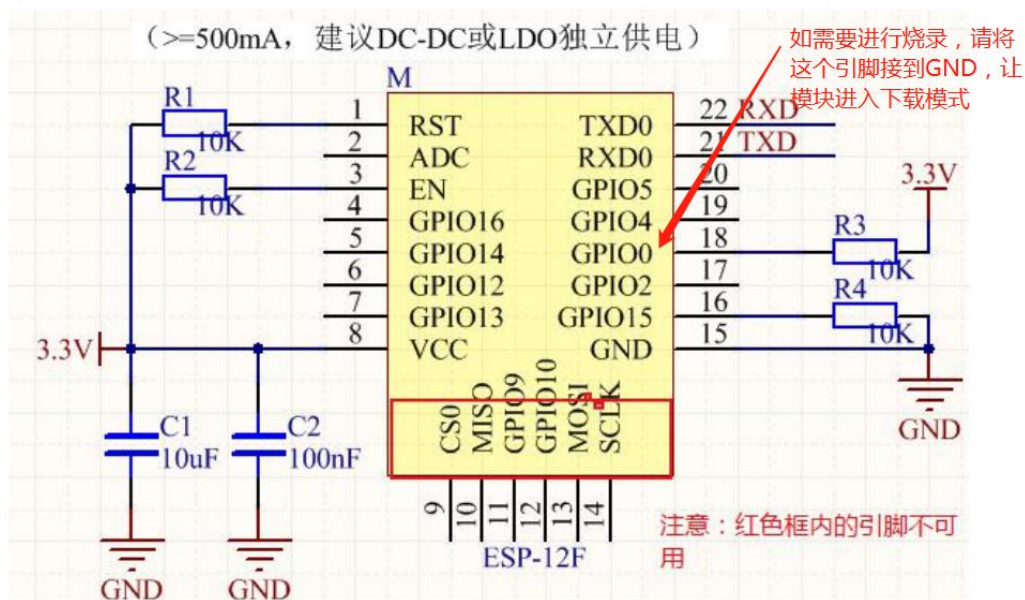
一、 烧录.....	5
1.1 ESP8266 模块如何进入烧录模式.....	5
1.2 如何正确使用 ESP8266 烧录软件对模块进行烧录.....	6
1.3 出现烧录失败的原因有哪些，该如何解决.....	8
二、 硬件.....	9
2.1 ESP-07、ESP-07S 使用的 IPEX 天线座子是第几代的.....	9
2.2 设计 ESP8266 的供电电路需要注意哪些问题.....	9
2.3 ESP8266 有多少种低功耗模式，每种低功耗模式的电流分别是多少.....	9
2.4 ESP8266 模组供电电压电流需求？.....	9
2.5 ESP-07 模块上能否直接外接天线，如不能外接需要做哪些处理？.....	10
2.6 ESP-12E, ESP-12F, ESP-12S, ESP-12L 这几款模块有什么区别.....	10
2.7 ESP8266 模组上电时电流很大是什么原因？.....	10
2.8 8266 模组上的 LED 是做什么使用的？为什么上电的时候闪烁一下就灭了？.....	10
2.9 为什么 8266 模组上电是输出乱码？.....	10
三、 应用.....	11
3.1 8266 模组通过串口发送 AT 指令无响应.....	11
3.2 如何查询模块的 MAC 地址.....	11
3.3 ESP8266 模块 AP 模式默认同时接入几个终端设备，最大能接入几个终端设备.....	11
3.4 8266TCP 包长和 UDP 包长为多少个字节.....	11
3.5 8266 为什么在透传过程中会出现丢包的情况.....	11
3.6 8266 使用 smartconfig 或者 airkiss 配不上网需要从哪方面检查原因.....	11
3.7 ESP8266NONSDK 和 RT0SSDK 有什么区别，在使用上有什么不同.....	11
3.8 ADC 的性能参数有几个通道？采样率和有效位数是多少？.....	12
3.9 ADC 的精度如何？.....	12
3.10 哪些 AT 指令会保存到 FLASH，上电之后不会丢失.....	13
3.11 关于 AT 指令的提升说明如下.....	13
3.12 如何屏蔽上电打印？.....	13
3.13 ESP8266 有几个 UART？.....	14
四、 SDK 开发.....	14
4.1 使用 AT 指令开发和使用 SDK 开发有什么优缺点.....	14
4.2 ESP8266 看门狗是什么作用.....	14
4.3 为什么 ESP8266 启动时打印 ets_main.c, 并且无法运行.....	14
4.4 ESP8266 的 RAM 的使用结构是怎样的.....	14
4.5 NONSDK 中哪些 API 接口会保存到 FLASH 中.....	14
4.6 为什么 ESP8266NONSDK 中有的函数前面添加了“ICACHE_FLASH_ATTR”宏？.....	14
4.7 如何通过 SDK 开发获取到模块的 MAC 地址.....	15
五、 联系我们.....	15

一、烧录

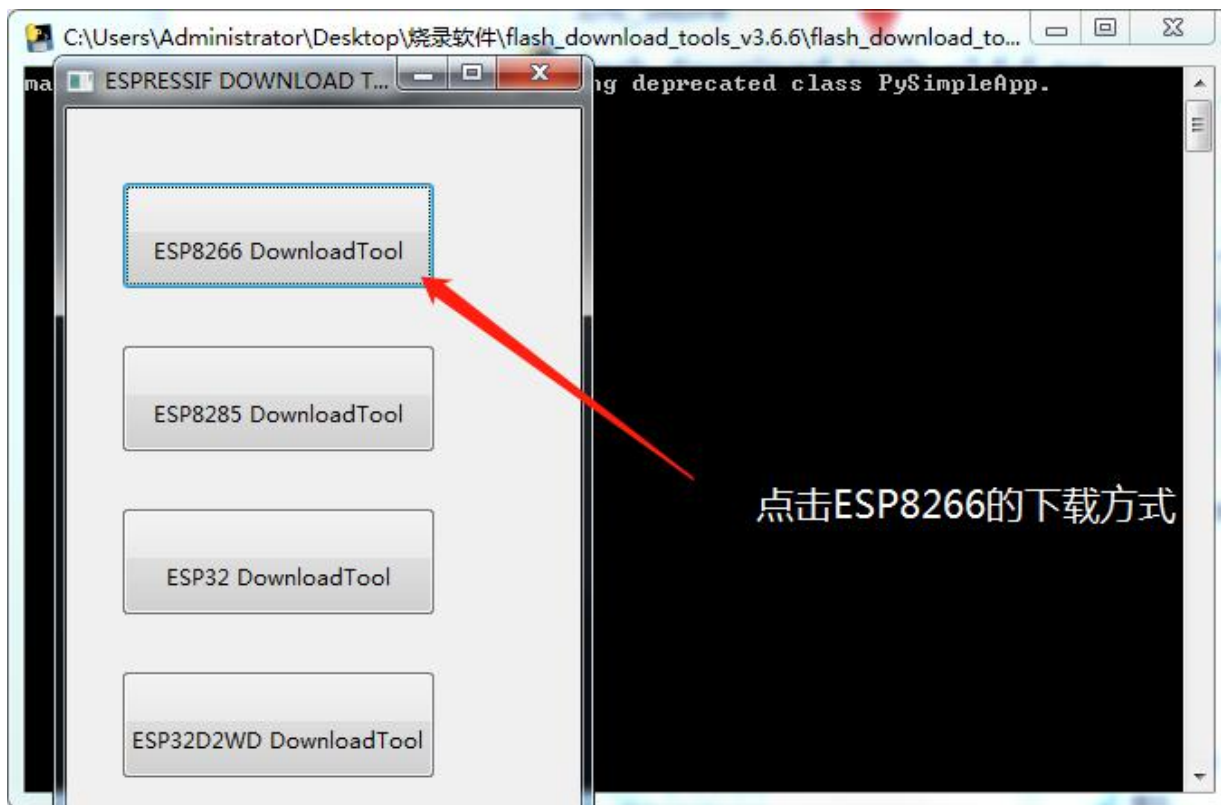
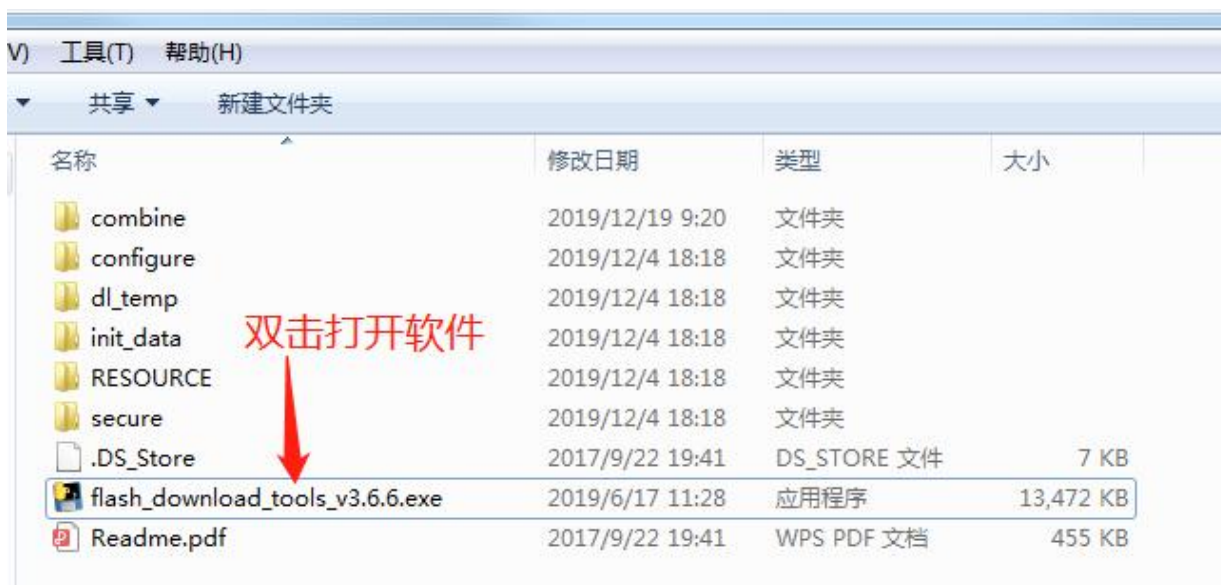
1.1 ESP8266 模块如何进入烧录模式

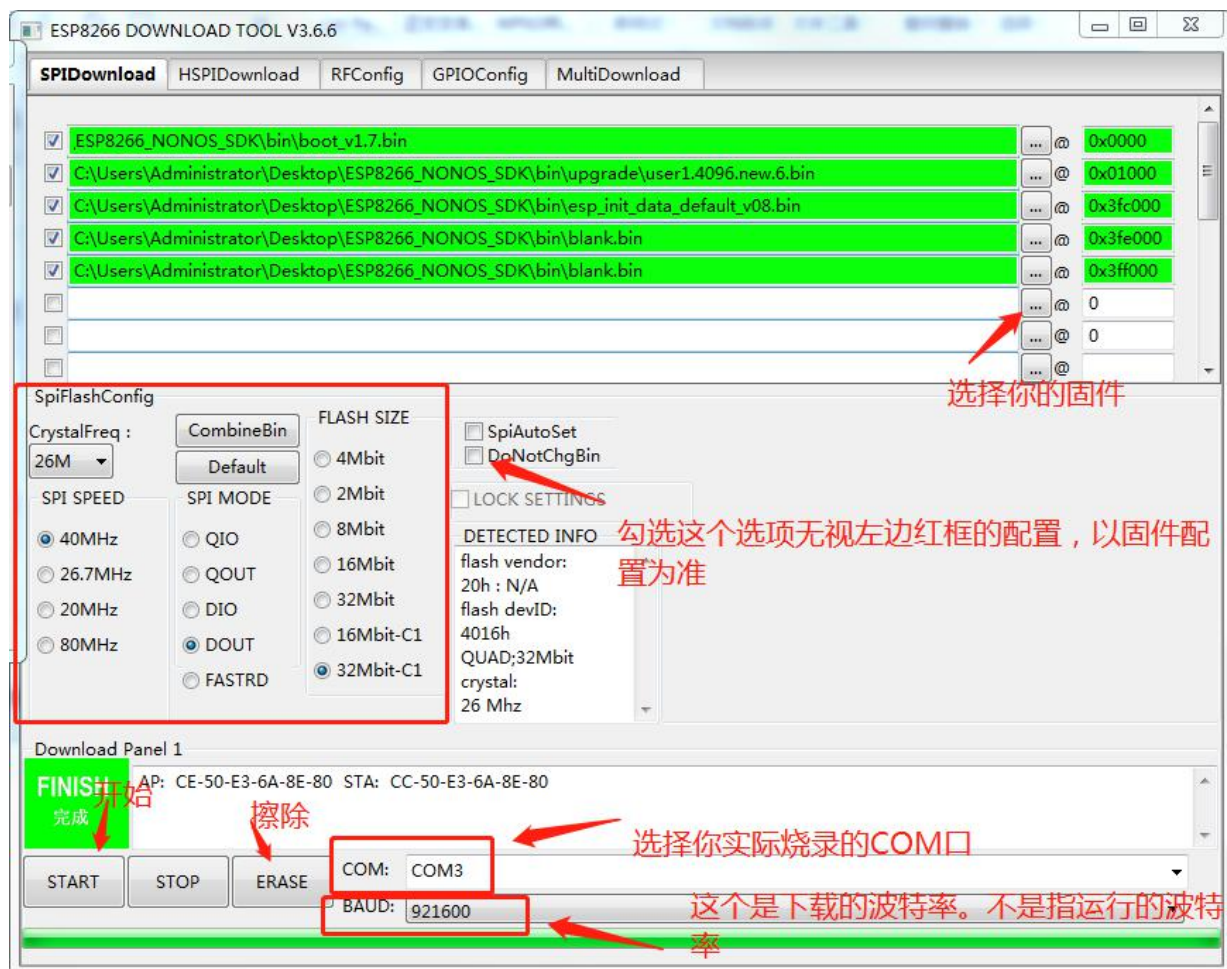
模块正常使用的应用电路基础上将模块的 GPIO0 引脚接 GND，模块重新上电读取波特率为 74880 的串口打印信息为 ets Jan 8 2013, rst cause:2, boot mode: (1, 6)，则代表模块已经成功进入烧录模式。这里以 ESP-12F 为参考，具体如图：

1、应用电路

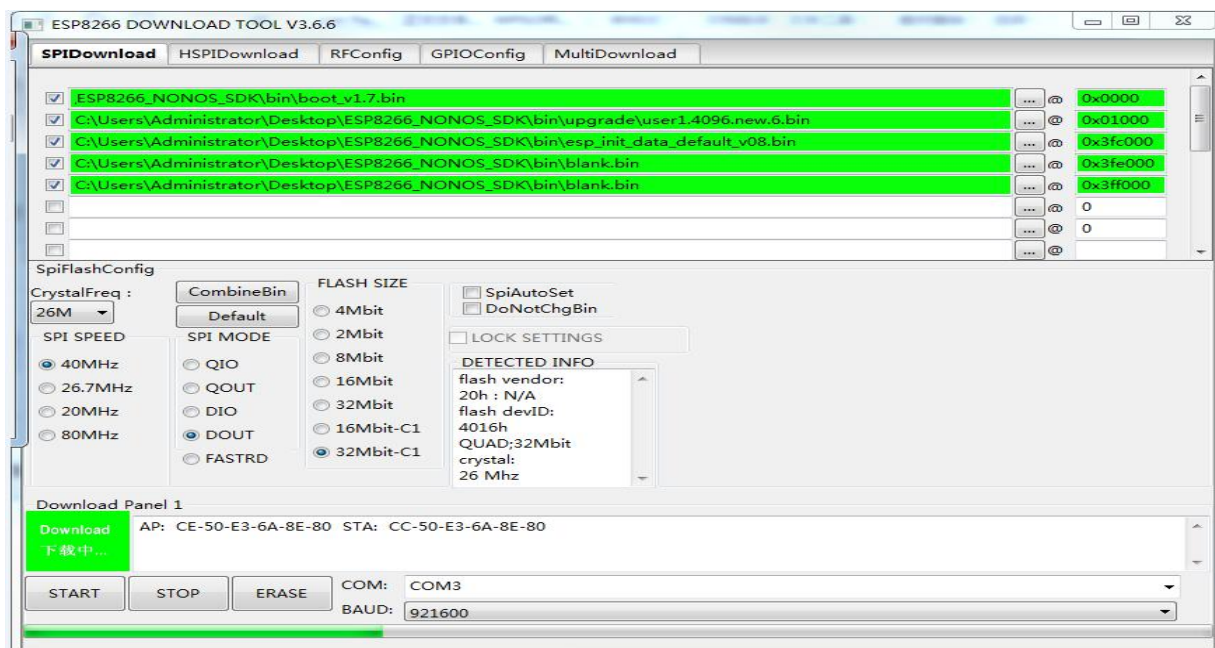


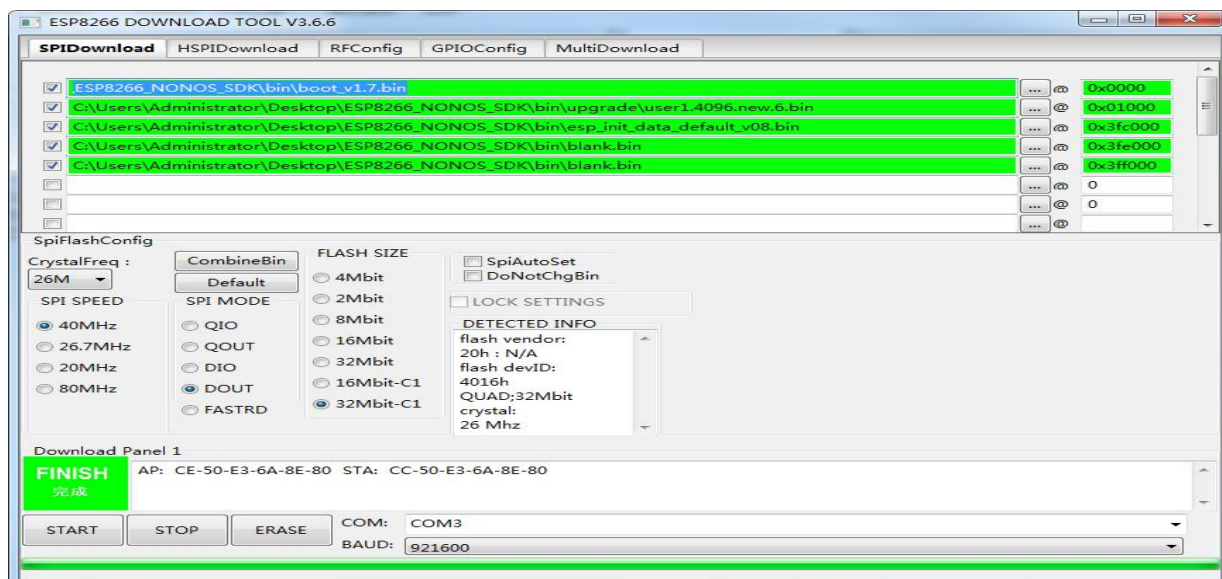
1.2 如何正确使用 ESP8266 烧录软件对模块进行烧录





如果出现如下图的情况代表模块已经开始烧录，软件下方会出现一段进度条，等进度条走完表示固件烧录完成，如下图所示：





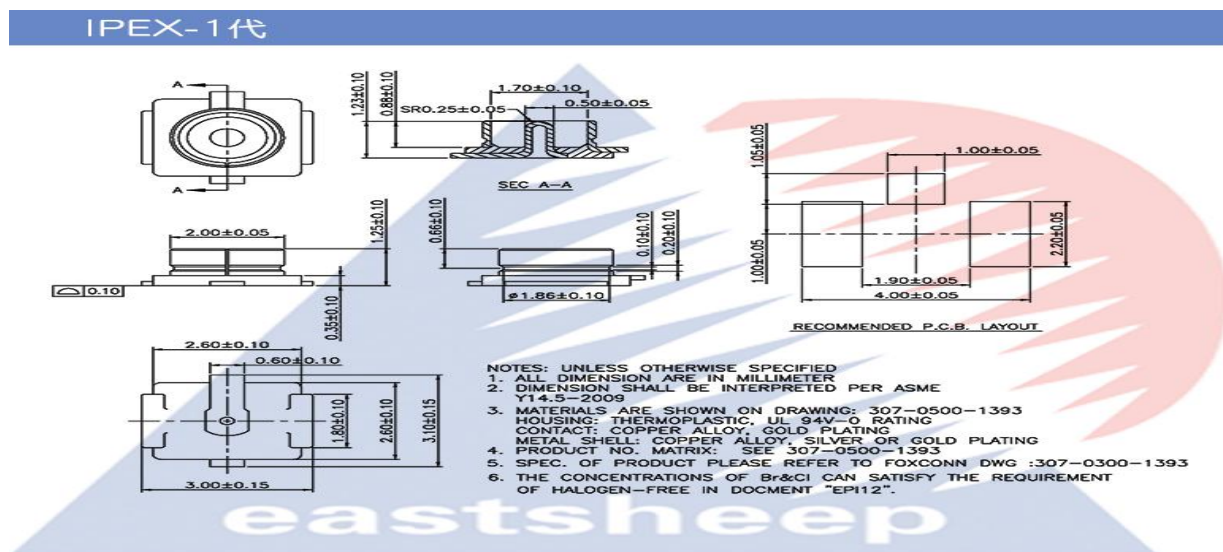
1.3 出现烧录失败的原因有哪些，该如何解决

- (1) COM 口选择错误或者 COM 被占用
- (2) 电源电压不稳定
- (3) 如果卡在等待上电过程，在确认接线无误的情况下将 RST 引脚接地复位下或者插拔一下 USB 转 TTL 的串口板
- (4) 串口芯片选型不对，模块的串口电平为 TTL 电平，串口芯片必须使用 CH340 或者 CP210X 的芯片，不能使用 232、485 甚至 PC 的九针孔接口来烧录
- (5) 串口不稳定，接入串口时，一定要把地线接上
- (6) 下载软件的 flash_size 选项超过了模组实际的 flash 大小，即 8Mbit 的 flash 按照 32M 烧录是不可以的
- (7) 下载波特率过大。有部分串口芯片不支持太高的波特率，甚至由于接线方式的原因、使用的烧录线品质较差、线太长等原因导致太高的波特率下载容易失败，需要适当降低下载波特率
- (8) efuse 损坏。由于静电的原因导致芯片损坏，下载软件的 efuse 校验无法通过。
- (9) 当 WIFI 模组的串口已经连接了用户产品的 MCU 串口，此时需要切断 MCU 与 WIFI 模组的串口连接或者将 MCU 的复位引脚拉低，让 MCU 处于复位模式

二、 硬件

2.1 ESP-07、ESP-07S 使用的 IPEX 天线座子是哪一代的

第一代，具体的规格可以参考如下图，可以使用游标卡尺进行测量



2.2 设计 ESP8266 的供电电路需要注意哪些问题

- (1) 如果使用 LDO 变压供电，请确保输入电压和输出电压足够大
- (2) 电源轨去耦电容器必须接近 ESP8266 摆放，等效电阻要足够低
- (3) ESP8266 不能直接连 5V 电压，如果需要使用 5V 供电请自行设计 5V 转 3V3 的电压转换
- (4) 如果使用 DC-DC 电压给 ESP8266 供电，必须要加上 LC 滤波电路

2.3 ESP8266 有多少种低功耗模式，每种低功耗模式的电流分别是多少

- (1) 模块正常运行时，电流约为 70mA
- (2) 模块进入 modem-sleep 模式下，电流约为 17.88mA
- (3) 模块进入 light-sleep 模式下，电流约为 2mA
- (4) 模块进入 deep-deep 模块下，电流约为 20uA

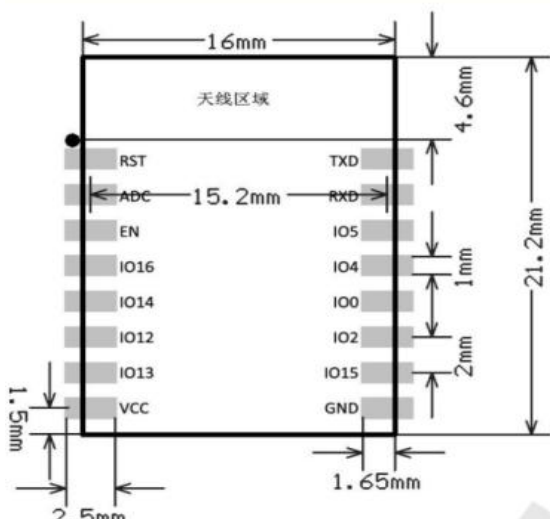
2.4 ESP8266 模组供电电压电流需求？

- (1) 推荐使用 3.3V，峰值 500mA 以上的电流进行供电以保证模组能正常工作
- (2) 建议使用 LDO 供电；如使用 DC-DC 建议纹波控制在 30mV 以内
- (3) DC-DC 供电电路建议预留动态响应电容的位置，可以在负载变化较大时，优化输出纹波
- (4) 3.3V 电源接口建议增加 ESD 器件

2.5 ESP-07 模块上能否直接外接天线，如不能外接需要做哪些处理？

ESP-07 模块不能直接外接天线，如果直接外接会导致天线的性能降低，如果需要外接天线需要将如图上的电阻去掉，断开与陶瓷天线的连接，如图所示

ESP-07 802.11 b/g/n Wi-Fi 模组



2.6 ESP-12E, ESP-12F, ESP-12S, ESP-12L 这几款模块有什么区别

ESP-12E 和 ESP-12L 使用的 PCB 板层为两层，ESP-12F 和 ESP-12S 使用的 PCB 板层为四层，在使用上比 ESP-12E 更加稳定，ESP-12S 和 ESP-12L 将 GPIO0、GPIO15、RST 和 EN 引脚进行了内部上下拉处理，模块接线只需要接 VCC、GND、TXD 和 RXD 即可使用，软件上没有任何的区别，可以相互替代

2.7 ESP8266 模组上电时电流很大是什么原因？

ESP8266 的 RF 和数字电路具有极高的集成度。上电后，RF 自校准会需要大电流。模拟部分电路最大的极限电路可能达到 500mA；数字电路部分最大电流达到 200mA。一般的操作，平均电流会在 100mA 左右。因此，ESP8266 需要供电达到 500mA 才能保证模块不会有瞬间压降

2.8 8266 模组上的 LED 是做什么使用的？为什么上电的时候闪烁一下就灭了？

(1) 因为 8266 模组的 LED 灯接到芯片的 GPIO2 引脚，此引脚与 UART1 的 TXD 共用，此时模块上电会输出一下打印信息，所以 LED 会闪烁直到打印信息输出完毕，LED 灯就灭了

(2) 若想查看启动信息的内容，请使用串口调试助手将波特率切换到 74880，硬件上吧 RST 引脚拉低再拉高可以查看启动信息

(3) 若想通过串口调试助手发送 AT 指令，请切换波特率到 115200 即可

2.9 为什么 8266 模组上电是输出乱码？

上电输出乱码是正常的，因为外部晶振选择是 26M，则 UART0 上电波特率， $26 \times 115200 / 40 = 74880$ ，请选择 74880 的波特率，按下复位键或把 RST 引脚拉低再拉高，可以看系统打印信息

三、应用

3.1 8266 模组通过串口发送 AT 指令无响应

- (1) 检查模组接线是否正确，可以参考我们规格书上的应用电路图
- (2) 串口助手设置是否在 115200 波特率，上电有无启动信息
- (3) 模块之前是否设置了一些上电自动进入透传的 AT 指令，如果设置请多次发送+++字符串给模块，让模块退出透传模式
- (4) 如果给模块发送 AT 指令，串口直接返回 AT 指令请检查是否加了\r\n，如果使用我司的串口调试助手查看发送新行的选项是否√，HEX 发送是否去掉√

3.2 如何查询模块的 MAC 地址

发送 AT+CIPSTAMAC_DEF? 进行查询，具体返回的参数如图

4.2.17. AT+CIPSTAMAC_DEF—设置 ESP8266 Station 当前 MAC 地址，保存到 Flash

指令	查询指令： AT+CIPSTAMAC_DEF?	设置指令： AT+CIPSTAMAC_DEF=<mac> 功能：设置 ESP8266 Station 的 MAC 地址。
响应	+CIPSTAMAC_DEF: <mac> OK	OK
参数说明	<mac>：字符串参数，ESP8266 Station 的 MAC 地址	
注意	<ul style="list-style-type: none"> 本设置保存到 Flash 用户参数区。 ESP8266 SoftAP 和 Station 的 MAC 地址并不相同，请勿将其设置为同一 MAC 地址。 ESP8266 MAC 地址第一个字节的 bit 0 不能为 1，例如，MAC 地址可以为 "18:..." 但不能为 "15:..."。 	
示例	AT+CIPSTAMAC_DEF="18:fe:35:98:d3:7b"	

3.3 ESP8266 模块 AP 模式默认同时接入几个终端设备，最大能接入几个终端设备

模块如果内部是 AT 固件，则默认同时接入 5 个终端设备，如果是使用 SDK 开发则最大能接入 8 个设备

3.4 8266TCP 包长和 UDP 包长为多少个字节

单包数据，TCP 包长 1460 个字节，UDP 包长 1472 个字节，一次性发送超过如上字节模块会自动分包发送

3.5 8266 为什么在透传过程中会出现丢包的情况

因为没有设置硬件流控。如果需要避免丢包，请设置硬件流控。透传功能使用的是 TCP 协议，每包数据是 1460（取决于协议栈），只要网络良好，buffer 空间没有被消耗完，就可以不停地传输数据。对于透传，串口接收数据间隔超过约 20 ms，就会认为数据接收结束，将已经接受的数据传输到网络。如果网络不好，就可能会丢弃一些数据，因此，为避免这种情况，可以将串口设置为流控模式。

3.6 8266 使用 smartconfig 或者 airkiss 配不上网需要从哪方面检查原因

- (1) 检查模块是否进入配网模式，具体可以点击我们的进行查看
- (2) 手机连接的路由器不能是单 5G 的路由器（双频除外）

3.7 ESP8266NONSDK 和 RTOS SDK 有什么区别，在使用上有什么不同

(1) Non-OS SDK 主要使用定时器和回调函数的方式实现各个功能事件的嵌套，达到特定条件下触发特定功能函数的目的。

(2) Non-OS SDK 使用 espconn 接口实现网络操作，用户需要按照 espconn 接口的使用规则进行软件开发。

(3) RTOS 版本 SDK 使用 freeRTOS 系统，引入 OS 多任务处理的机制，用户可以使用

freeRTOS 的标准接口实现资源管理、循环操作、任务内延时、任务间信息传递和同步等面向任务流程的设计方式。

(4) RTOS 版本 SDK 的网络操作接口是标准 lwIP API，同时提供了 BSD Socket API 接口的封装实现，用户可以直接按照 socket API 的使用方式来开发软件应用，也可以直接编译运行其他平台的标准 Socket 应用，有效降低平台切换的学习成本。

(5) RTOS 版本 SDK 引入了 cJSON 库，使用该库函数可以更加方便的实现对 JSON 数据包的解析。

(6) RTOS 版本兼容 Non-OS SDK 中的 Wi-Fi 接口、SmartConfig 接口、Sniffer 相关接口、系统接口、定时器接口、FOTA 接口和外围驱动接口，不支持 AT 实现。

3.8 ADC 的性能参数有几个通道？采样率和有效位数是多少？

通道：1

采样率：

- 停止 Wi-Fi 的情况下，能达到每秒 100000 次。
- Wi-Fi 正常工作的情况下，能达到每秒 1000 次。

有效位数：

内部 ADC 有效位数为 12 位。

`system_adc_read()` API 返回值的有效位数是 10 位。

3.9 ADC 的精度如何？

ESP8266 连接路由器后，单 STA 模式会进入 modem-sleep，导致芯片内部电流发生变化，参考值变化，因此 ADC 采集异常。

用户如果需要测量的非常准确，可以用 `system_adc_fast_read` 的函数，但是测量之前需

要关闭 RF，Wi-Fi 连接会断开。如果需要测试比较准确，数值相差 1，或 2，可以配置

Wi-Fi 为 non-sleep 模式 `wifi_set_sleep_type(NONE_SLEEP_T)`；建议该用户这样配置。

如果对精确性要求不高，可以允许模块进入 sleep 模式，功耗较低。

3.10 哪些 AT 指令会保存到 FLASH，上电之后不会丢失

A.

附录 A

以下 ESP8266 AT 指令会保存设置到 Flash:

指令	示例
保存在 Flash 用户参数区	
AT+UART_DEF	AT+UART_DEF=115200,8,1,0,3
AT+CWDHCP_DEF	AT+CWDHCP_DEF=1,1
AT+CIPSTAMAC_DEF	AT+CIPSTAMAC_DEF="18:fe:35:98:d3:7b"
AT+CIPAPMAC_DEF	AT+CIPAPMAC_DEF="1a:fe:36:97:d5:7b"
AT+CIPSTA_DEF	AT+CIPSTA_DEF="192.168.6.100"
AT+CIPAP_DEF	AT+CIPAP_DEF="192.168.5.1"
AT+CWDHCPS_DEF	AT+CWDHCPS_DEF=1,3,"192.168.4.10","192.168.4.15"
AT+SAVETRANSLINK	AT+SAVETRANSLINK_DEF=1,"192.168.6.10",1001
AT+CIPDNS_DEF	AT+CIPDNS_DEF=1,"208.67.220.220"
AT+SYSMSG_DEF	AT+SYSMSG_DEF=3
AT+CWCOUNTRY_DEF	AT+CWCOUNTRY_DEF=1,"CN",1,5
AT+CIPSSLCONF	AT+CIPSSLCONF=2
保存在 Flash 系统参数区	
AT+CWMODE_DEF	AT+CWMODE_DEF=3
AT+CWJAP_DEF	AT+CWJAP_DEF="abc","0123456789"
AT+CWSAP_DEF	AT+CWSAP_DEF="ESP8266","12345678",5,3
AT+CWAUTOCONN	AT+CWAUTOCONN=1

注意:

- 以上指令设置时，会先读取 Flash 中的原配置，仅新配置与原配置不同时，才写 Flash 保存新配置。
- 对于 512 KB + 512 KB Flash Map，用户参数区为 0x7C000 ~ 0x80000，16 KB；
- 对于 1024 KB + 1024 KB Flash Map：用户参数区为 0xFC000 ~ 0x100000，16 KB；
- 系统参数区始终为 Flash 的最后 16 KB。

3.11 关于 AT 指令的提升说明如下

B.

附录 B

ESP8266 AT 指令中的提示信息说明如下：

提示信息	说明
ready	AT 固件成功启动
ERROR	指令输入错误，或者指令执行出错
WIFI CONNECTED	ESP8266 station 连接到 AP
WIFI GOT IP	ESP8266 station 获取到 IP 地址
WIFI DISCONNECT	ESP8266 station 的 WiFi 连接断开
busy s...	busy sending，表示系统正在发送数据的过程中，无法响应当前输入
busy p...	busy processing，表示系统正在处理前一条指令，无法响应当前输入
<conn_id>,CONNECT	建立了 <conn_id> 号网络连接
<conn_id>,CLOSED	<conn_id> 号网络连接断开
+IPD	接收到网络数据
+STA_CONNECTED:<sta_mac>	有 station 连入 ESP8266 softAP
+DIST_STA_IP:<sta_mac>,<sta_ip>	ESP8266 softAP 给连入的 station 分配 IP 地址
+STA_DISCONNECTED:<sta_mac>	station 从 ESP8266 softAP 断开连接

3.12 ESP8266 如何屏蔽上电打印？

U0TXD 默认上电有系统打印，对此敏感应用可通过 UART 的内部引脚交换功能，在初始化的时候，

调用 `system_uart_swap` 函数，将 U0TXD、U0RXD 分别与 U0RTS（MTD0/

GPI015），U0CTS（MTCK/GPI013）交换来屏蔽该上电的系统打印。

交换后，硬件上的下载管脚还是使用 U0TXD + U0RXD，通信时需要将 MTD0 对应接到

MCU 的 RXD，MTCK 对应接到 MCU 的 TXD。

3.13 ESP8266 有几个 UART?

ESP8266 有两个 UART，其中 UART0 有 TX、RX，可做数据传输；UART1 由于 RX 脚被 SPI-Flash 占用，只能使用 TX，可以做串口调试信息打印。

四、SDK 开发

4.1 使用 AT 指令开发和使用 SDK 开发有什么优缺点

AT 指令开发：只需要知道几条 AT 指令即可用外部单片机实现网络通讯，开发速度快，但是增加了外置 MCU 的成本

SDK 开发：让系统成本最小，体积更小，能实现更多的功能，但是入门门槛较高，新手需要花一定的时间去熟悉代码的研读

4.2 ESP8266 看门狗是什么作用

为了提供系统的稳定性，以应对多冲突的操作环境，ESP8266 集成了 2 级看门狗机制，包括软件看门狗和硬件看门狗，默认两个看门狗都是打开的

4.3 为什么 ESP8266 启动时打印 ets_main.c，并且无法运行

启动时打印 ets_main.c 表示没有可运行的程序区，无法运行；遇到这种问题时，请检查烧录时的 bin 文件和烧录地址是否正确

4.4 ESP8266 的 RAM 的使用结构是怎样的

ESP8266 的 RAM 总共 160 KB。

- IRAM 空间为 64 KB:
前 32 KB 用作 IRAM，用来存放没有加 `ICACHE_FLASH_ATTR` 的代码，即 .text 段，会通过 ROM code 或二级 boot 从 SPI Flash 中的 BIN 中加载到 IRAM；
后 32 KB 被映射作为 iCache，放在 SPI Flash 中的，加了 `ICACHE_FLASH_ATTR` 的代码会被从 SPI Flash 自动动态加载到 iCache。
- DRAM 空间为 96 KB:
对于 Non-OS_SDK，前 80 KB 用来存放 `.data/.bss/.rodata/heap`，heap 区的大小取决于 `.data/.bss/.rodata` 的大小；还有 16 KB 给 ROM code 使用。
对于 RTOS_SDK，96 KB 用来存放 `.data/.bss/.rodata/heap`，heap 区的大小取决于 `.data/.bss/.rodata` 的大小。

4.5 NONSDK 中哪些 API 接口会保存到 FLASH 中

```
wifi_station_set_auto_connect
wifi_station_ap_number_set
wifi_set_phy_mode
wifi_softap_set_config
wifi_station_set_config
wifi_set_opmode
system_restart_enhance
system_restore
system_upgrade_reboot
```

4.6 为什么 ESP8266 NONSDK 中有的函数前面添加了“ICACHE_FLASH_ATTR”宏?

对于 ESP8266_Non-OS_SDK:

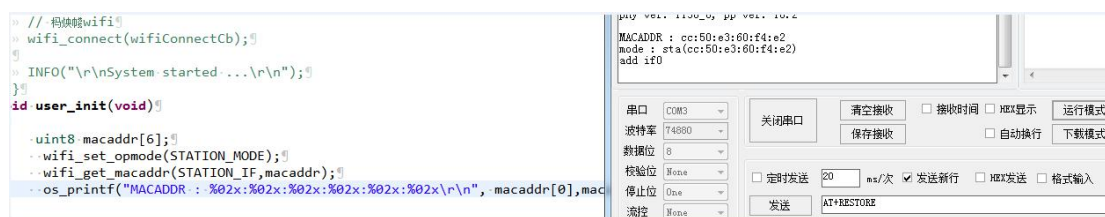
添加了“**ICACHE_FLASH_ATTR**”宏的函数，将存放在 IROM 中，CPU 仅在调用到它们的时候，将它们读到 cache 中运行；没有添加“**ICACHE_FLASH_ATTR**”宏的函数，将在一开始上电运行时，就加载到 IRAM 中运行；由于空间有限，我们无法将所有代码都一次性加载到 IRAM 中运行，因此在大部分函数前添加“**ICACHE_FLASH_ATTR**”宏，放在 IROM 中。

请注意，不要在中断处理函数中调用带有“**ICACHE_FLASH_ATTR**”宏的函数，否则可能与 Flash 读写操作冲突。

对于 ESP8266_RTOS_SDK:

函数默认存放在 IROM 中，无需再添加“**ICACHE_FLASH_ATTR**”宏。中断处理函数也可以定义在 IROM 中。如果开发者需要将一些频繁调用的函数定义在 IRAM 中，在函数前添加“**IRAM_ATTR**”宏即可。

4.7 如何通过 SDK 开发获取到模块的 MAC 地址



五、 联系我们

官方官网: <https://www.ai-thinker.com>

开发 DOCS: <http://docs.aithinker.com>

样品购买: <https://anxinke.taobao.com>

商务合作: sales@aitinker.com

技术支持: support@aitinker.com

公司地址: 深圳市宝安区西乡固戍华丰智慧创新港 C 栋 410

联系电话: 0755-29162996

